

ppedv

TSQL – Labs - Angaben

Übungen für TSQL Einsteiger
Kurs SQL Abfragesprache und Datenbankdesign

TSQL – Labs – Angaben

Inhalt

Getting Started	3
Datenbankdiagramm Northwind-DB	5
Modul 1 – Einfache Abfragen	6
Übung 1.1	6
Übung 1.2	6
Übung 1.3	7
Übung 1.4	7
Modul 2 – String- und Datumsfunktionen	8
Übung 2.1	8
Übung 2.2	8
Übung 2.3	9
Übung 2.4	9
Übung 2.5	10
Übung 2.6	10
Modul 03 – WHERE-Klausel	11
Übung 3.1	11
Übung 3.2	11
Übung 3.3	11
Übung 3.4	12
Übung 3.5	12
Übung 3.6	13
Übung 3.7	13
Übung 3.8	13
Modul 04 – Wildcards	15
Übung 4.1	15
Übung 4.2	15
Übung 4.3	16
Modul 05 – JOINS	17
Übung 5.1	17
Übung 5.2	18
Übung 5.3	18

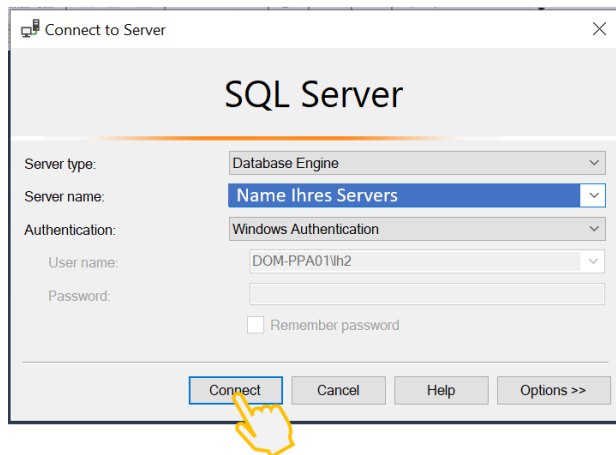


Übung 5.4.....	19
Übung 5.5.....	19
Übung 5.6.....	20
Übung 5.7.....	20
Modul 06 – DISTINCT, TOP, ORDER BY.....	22
Übung 6.1.....	22
Übung 6.2.....	22
Übung 6.3.....	23
Modul 07 – Aggregatfunktionen; HAVING.....	24
Übung 7.1.....	24
Übung 7.2.....	24
Übung 7.3.....	25
Übung 7.4.....	25
Übung 7.5.....	26
Übung 7.6.....	26
Übung 7.7.....	27
Modul 08 – temporäre Tabellen, Views (Sichten), Prozeduren.....	28
Übung 8.1.....	28
Übung 8.2.....	29
Übung 8.3.....	29
Modul 09 – SET-Operatoren.....	31
Übung 9.1.....	31
Übung 9.2.....	32
Modul 10 – Subqueries (Unterabfragen).....	33
Übung 10.1.....	33
Übung 10.2.....	33
Übung 10.3.....	34
Übung 10.4.....	34
Übung 10.5.....	35
Übung 10.6.....	35
Modul 11 – CASE.....	37
Übung 11.1.....	37
Übung 11.2.....	37
Modul 12 - CREATE, INSERT, UPDATE, DELETE.....	39
Übung 12.1.....	39

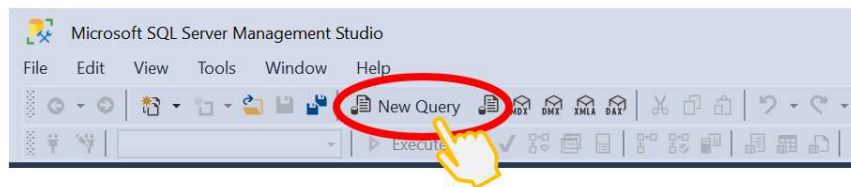
Getting Started

Für die Übungen benötigen Sie eine beliebige Version von SQL-Server und SSMS (Microsoft SQL Server Management Studio).

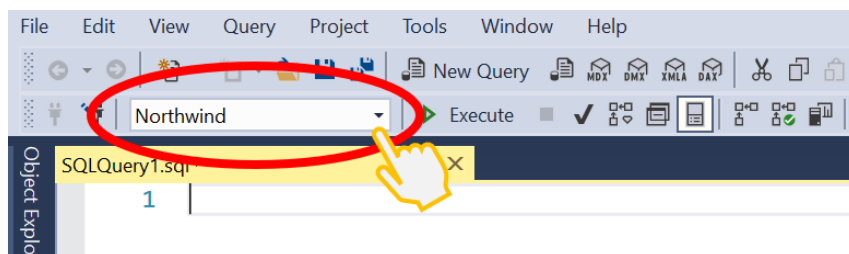
Verbinden Sie sich mit Ihrem Server:



Öffnen Sie ein neues Abfragefenster mit STRG+N oder indem Sie in der Taskleiste auf „New Query“ klicken:

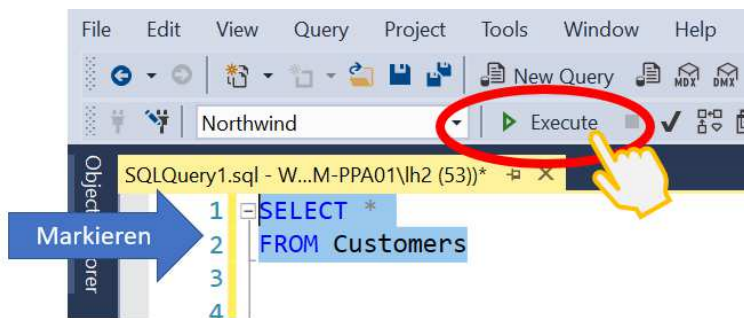


Stellen Sie sicher, dass Sie die Übungsdatenbank Northwind verwenden:



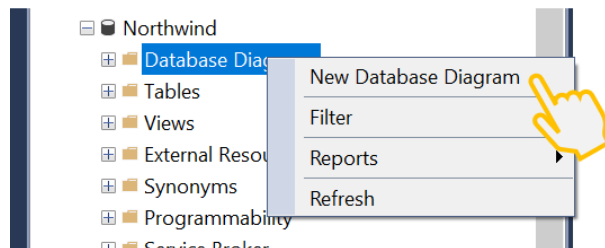
Jetzt können Sie Ihre Abfragen schreiben.

Um Ihre Abfragen auszuführen, markieren Sie den auszuführenden Code und klicken Sie auf „Execute“ (oder drücken Sie F5):

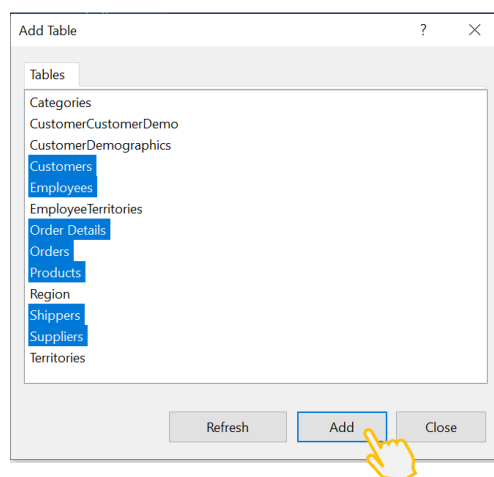


Wenn Sie sich in der Übungsdatenbank Northwind erst zurechtfinden müssen, können Sie zur besseren Übersichtlichkeit ein Datenbankdiagramm erstellen:

Im Object-Explorer (standardmäßig links angedockt, oder öffnen mit F8) klicken Sie auf die gewünschte Datenbank, rechtsklicken auf „Database Diagram“ und wählen „New Database Diagram“.



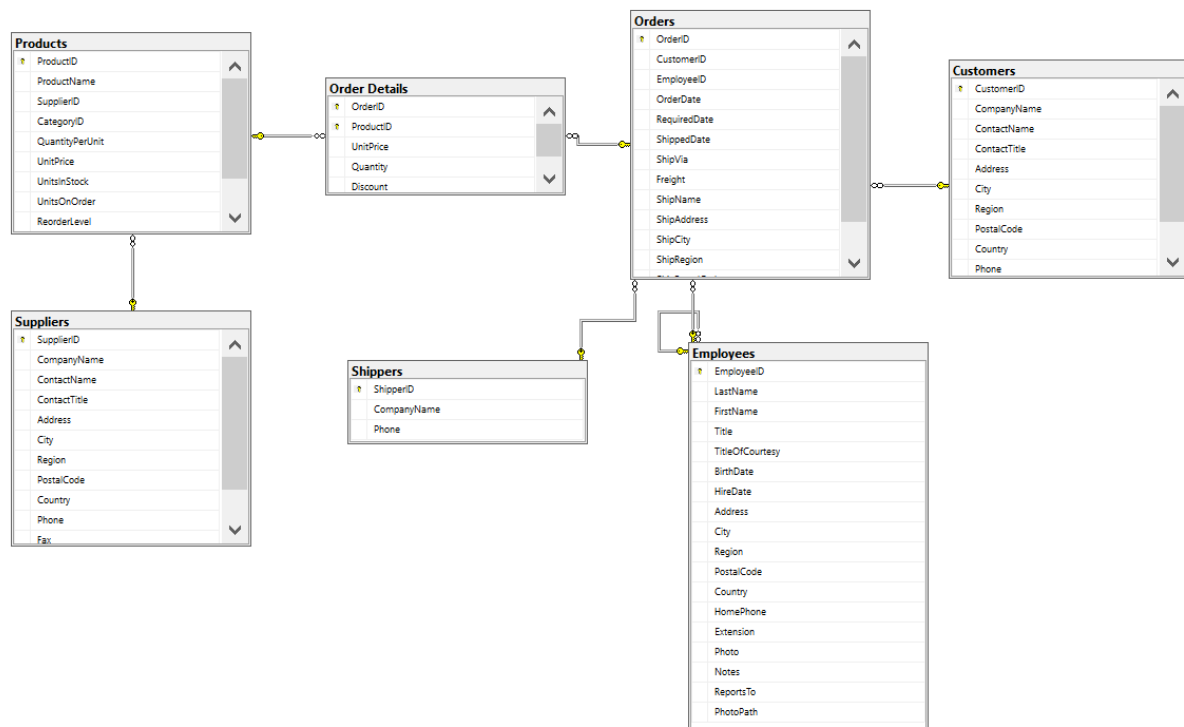
Hier können Sie alle Tabellen auswählen, die im Datenbankdiagramm angezeigt werden sollen. Für unsere Übungen reichen uns die Tabellen Customers, Employees, Order Details, Orders, Products, Shippers und Suppliers:



Mit Klick auf Add fügen Sie diese Tabellen ihrem Datenbankdiagramm hinzu. Jetzt können Sie das „Add Table“-Fenster schließen und Ihre Tabellen nach Belieben anordnen. Wenn Sie eine Tabelle zu viel oder zu wenig ausgewählt haben, können Sie mit Rechtsklick das „Add Table“-Fenster wieder öffnen oder Tabellen aus dem Diagramm entfernen.

Hier gibt es einen kleinen Überblick über die für die Übungen verwendeten Tabellen in der Northwind-Übungsdatenbank:

Datenbankdiagramm Northwind-DB



Modul 1 – Einfache Abfragen

Übung 1.1

Ziel: Textausgabe mit eigener Spaltenüberschrift erzeugen

Aufgabenstellung:

Erzeugen Sie folgende Ausgabe:

	Zahl	Text	Rechnung
1	100	Donaudampfschiffahrtsgesellschaft	200

Die Ausgabe bei „Rechnung“ soll sich aus einer *Berechnung* im SELECT ergeben (z.B.: 100*2).

TIPPS:

Ein ALIAS (im Grunde ist damit eine Spaltenüberschrift in der Textausgabe gemeint) wird mit „AS“ erstellt.

Syntax:

SELECT Spaltenname AS Spaltenüberschrift

Übung 1.2

Ziel: Üben von einfachen Tabellenabfragen und Kennenlernen der Northwind-Übungsdatenbank

Aufgabenstellung: Geben Sie KundenID, Firmenname, Kontaktperson und Telefonnummer aller Kunden aus.

	KundenID	Firmenname	Kontaktperson	Telefonnummer
1	ALFKI	Alfreds Futterkiste	Maria Anders	030-0074321
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	(5) 555-4729
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	(5) 555-3932
4	AROUT	Around the Horn	Thomas Hardy	(171) 555-7788
5	BERGS	Berglunds snabbköp	Christina Berglund	0921-12 34 65
6	BLAUS	Blauer See Delikatessen	Hanna Moos	0621-08460

Ergebnis: 91 Zeilen (alle Kunden)

Übung 1.3

Ziel: Üben von einfachen Tabellenabfragen und Kennenlernen der Northwind-Übungsdatenbank

Aufgabenstellung: Geben Sie die ProduktID, den Produktnamen und den Stückpreis aus.

	ProduktID	Produktname	Stückpreis
1	1	Chai	18,00
2	2	Chang	19,00
3	3	Aniseed Syrup	10,00
4	4	Chef Anton's Cajun Seasoning	22,00
5	5	Chef Anton's Gumbo Mix	21,35

Ergebnis: 77 Zeilen (alle Produkte)

Übung 1.4

Ziel: Üben von einfachen Tabellenabfragen und Anwenden von Berechnungen im SELECT

Aufgabenstellung: Geben Sie die Nettofrachtkosten, Bruttofrachtkosten und die Mehrwertsteuer aus (Berechnung).

Annahme: Die Frachtkosten (Freight) in den Bestellungen (Orders) sind Nettofrachtkosten und die MwSt beträgt 19%.

	Nettofrachtkosten	Bruttofrachtkosten	MwSt
1	32,38	38.532200	6.152200
2	11,61	13.815900	2.205900
3	65,83	78.337700	12.507700
4	41,34	49.194600	7.854600
5	51,30	61.047000	9.747000

Ergebnis: 830 Zeilen

Modul 2 – String- und Datumsfunktionen

Übung 2.1

Ziel: Üben von String Funktionen

Aufgabenstellung: Die letzten drei Stellen einer Telefonnummer sollen durch xxx ersetzt werden.

Bonus: die Lösung soll unabhängig von der Länge der Telefonnummer funktionieren

TIPPS:

- Beginnen Sie zunächst mit der Zahlenfolge 1234567890. Wenn es damit funktioniert, versuchen Sie, die Anzahl der Zeichen zu verändern und die Abfrage so umzuformulieren, dass sie unabhängig von der Anzahl der Ziffern funktioniert. Testen Sie es an den Telefonnummern der Kunden in der Customers-Tabelle.
- Serverfunktionen können miteinander verschachtelt werden.
- Es gibt mehrere Möglichkeiten, diese Aufgabe umzusetzen. Hilfreiche Funktionen für diese Aufgabenstellung sind z.B. CONCAT, LEFT, LEN, STUFF und/oder REVERSE.

	SecretPhone	Phone	CompanyName
1	030-0074xxx	030-0074321	Alfreds Futterkiste
2	(5) 555-4xxx	(5) 555-4729	Ana Trujillo Emparedados y helados
3	(5) 555-3xxx	(5) 555-3932	Antonio Moreno Taquería
4	(171) 555-7xxx	(171) 555-7788	Around the Horn
5	0921-12 34xxx	0921-12 34 65	Berglunds snabbköp

Ergebnis: 91 Zeilen (alle Kunden)

Übung 2.2

Ziel: Üben von String Funktionen

Aufgabenstellung: Umgekehrt als in der Übung eben, sollen von der Telefonnummer aus der Customers-Tabelle nur die letzten 3 Zeichen angezeigt werden; alle anderen sollen mit x ersetzt werden.

(xxxxxxxxxxxxxx890)

	PhoneX	Phone
1	xxxxxxxx321	030-0074321
2	xxxxxxxx729	(5) 555-4729
3	xxxxxxxx932	(5) 555-3932
4	xxxxxxxx788	(171) 555-7788
5	xxxxxxxx 65	0921-12 34 65

Ergebnis: 91 Zeilen (alle Kunden)

Übung 2.3

Ziel: Üben von String Funktionen

Aufgabenstellung: Geben Sie an, an welcher Stelle sich das letzte Leerzeichen befindet. Testen Sie z.B. mit „Wolfgang Amadeus Mozart“ und „Georg Friedrich Händel“.

TIPP: Programmierer müssen umdenken! In SQL beginnen wir bei 1, nicht bei 0 zu zählen. Das „W“ in Wolfgang befindet sich an der 1. (ersten), NICHT an der 0. („nullten“) Stelle wie in einer Programmiersprache.

Results	Messages
	letztes Leerzeichen
1	17

Ergebnis: Mozart 17, Händel 16

Übung 2.4

Ziel: Üben von Datumsfunktionen

Aufgabenstellung:

- Finden Sie mittels Datumsfunktion heraus, an welchem Wochentag Sie geboren wurden.
- Wenden Sie die Funktion auf die Geburtsdaten in der Northwind-Datenbank an, um zu sehen, an welchen Wochentagen die Angestellten geboren wurden. Neben dem Wochentag soll auch der Name der Angestellten (Vor- und Nachname in einem Feld) stehen.

TIPPS: Datenbankdiagramm verwenden (s .o.)

Results	Messages
	Leo's Birthday
1	Wednesday

Results	Messages
	Emp Birthday
1	Wednesday
2	Tuesday
3	Friday
4	Sunday
5	Friday

Das Ergebnis soll der ausgeschriebene Name des Wochentages sein, an dem Sie (bzw. die Angestellten) geboren wurden. (Das Ergebnis wird natürlich immer unterschiedlich ausfallen.)

In der Northwind-DB gibt es insgesamt 9 Angestellte.

Übung 2.5

Ziel: Üben von Datumsfunktionen

Aufgabenstellung: Vor wie vielen Jahren kam der erste Star Wars Film in die Kinos? (25. Mai 1977)

Verwenden Sie das aktuelle Datum.

Results		Messages	
		Movie	
1	44		

Übung 2.6

Ziel: Üben von Datums- und anderen Serverfunktionen

Aufgabenstellung:

- Berechnen Sie das Alter der Angestellten in der Northwind-DB. Beachten Sie, dass DATEDIFF nur die Differenz zwischen dem angegebenen Intervall berücksichtigt.
- Geben Sie Geburtsdatum, Alter und Name (Vor- und Nachname in einer Spalte) der Angestellten aus.
- Verwenden Sie das aktuelle Datum.

Results		Messages	
	BirthDate	EmpAge	EmpName
1	08.12.1948	72	Nancy Davolio
2	19.02.1952	69	Andrew Fuller
3	30.08.1963	57	Janet Leverling
4	19.09.1937	83	Margaret Peacock
5	04.03.1955	65	Steven Buchanan

Hinweise:

- Insgesamt gibt es 9 Angestellte.
- Das Alter kann vom Bild oben abweichen (abhängig vom aktuellen Datum).

Modul 03 – WHERE-Klausel

Übung 3.1

Ziel: Einüben von einfachen Bedingungen im WHERE

Aufgabenstellung: Geben Sie die Kontaktdaten der Kunden aus, die in Frankreich ansässig sind.

	CustomerID	CompanyName	ContactName	Phone
1	BLONP	Blondesddsl père et fils	Frédérique Citeaux	88.60.15.31
2	BONAP	Bon app'	Laurence Lebihan	91.24.45.40
3	DUMON	Du monde entier	Janine Labrune	40.67.88.88
4	FOLIG	Folies gourmandes	Martine Rancé	20.16.10.16
5	FRANR	France restauration	Carine Schmitt	40.32.21.21

Ergebnis: 11 Zeilen

Übung 3.2

Ziel: Einüben von einfachen Bedingungen im WHERE

Aufgabenstellung: Geben Sie die Kontaktdaten der Kunden aus, die in Buenos Aires in Argentinien ansässig sind.

	CustomerID	CompanyName	ContactName	Phone
1	CACTU	Cactus Comidas para llevar	Patricio Simpson	(1) 135-5555
2	OCEAN	Océano Atlántico Ltda.	Yvonne Moncada	(1) 135-5333
3	RANCH	Rancho grande	Sergio Gutiérrez	(1) 123-5555

Ergebnis: 3 Zeilen

Übung 3.3

Ziel: Einüben von einfachen Bedingungen im WHERE

Aufgabenstellung: Geben Sie die Kontaktdaten aller portugiesischen und spanischen Kunden aus.

	CustomerID	CompanyName	ContactName	Phone
1	BOLID	Bólido Comidas preparadas	Martín Sommer	(91) 555 22 82
2	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	(91) 555 94 44
3	FURIB	Furia Bacalhau e Frutos do Mar	Lino Rodriguez	(1) 354-2534
4	GALED	Galería del gastrónomo	Eduardo Saavedra	(93) 203 4560
5	GODOS	Godos Cocina Típica	José Pedro Freyre	(95) 555 82 82

Ergebnis: 7 Zeilen

Übung 3.4

Ziel: Einüben von einfachen Bedingungen im WHERE

Aufgabenstellung: Geben Sie alle Produkte aus, von denen mehr als 100 vorhanden sind.

TIPP: Wir müssen hier nichts berechnen, das steht in der Datenbank.

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25,00	120	0	25	0
2	22	Gustaf's Knäckebröd	9	5	24 - 500 g pkgs.	21,00	104	0	25	0
3	33	Geitost	15	4	500 g	2,50	112	0	20	0
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	14,00	111	0	15	0
5	36	Inlagd Sill	17	8	24 - 250 g jars	19,00	112	0	20	0

Ergebnis: 10 Zeilen

Übung 3.5

Ziel: Einüben von einfachen Bedingungen im WHERE, Verwendung von BETWEEN

Aufgabenstellung:

- Geben Sie alle Produkte aus, deren ProduktID zwischen 10 und 15 (inklusive) liegt.
- Wählen Sie passende Spalten aus.

	ProductID	ProductName	UnitPrice
1	10	Ikura	31,00
2	11	Queso Cabrales	21,00
3	12	Queso Manchego La Pastora	38,00
4	13	Konbu	6,00
5	14	Tofu	23,25
6	15	Genen Shouyu	15,50

Ergebnis: 6 Zeilen

Übung 3.6

Ziel: Einüben von einfachen Bedingungen im WHERE, Verwendung von IN

Aufgabenstellung:

- Geben Sie alle Produkte aus, die von den Anbietern (SupplierID) 2, 7 oder 15 geliefert werden.
- Wählen Sie passende Spalten aus.

	ProductID	ProductName	UnitPrice
1	4	Chef Anton's Cajun Seasoning	22,00
2	5	Chef Anton's Gumbo Mix	21,35
3	16	Pavlova	17,45
4	17	Alice Mutton	39,00
5	18	Carnarvon Tigers	62,50

Ergebnis: 12 Zeilen

Übung 3.7

Ziel: Einüben von einfachen Bedingungen im WHERE

Aufgabenstellung:

- Geben Sie alle Produkte aus, die vom Anbieter (SupplierID) 5, 10 oder 15 stammen, von denen mehr als 10 Stück vorrätig sind und deren Stückpreis unter 100 liegt.
- Wählen Sie passende Spalten aus.

	ProductID	ProductName	UnitPrice	UnitsInStock
1	11	Queso Cabrales	21,00	22
2	12	Queso Manchego La Pastora	38,00	86
3	24	Guaraná Fantástica	4,50	20
4	33	Geitost	2,50	112
5	69	Gudbrandsdalsost	36,00	26
6	71	Flotemysost	21,50	26

Ergebnis: 6 Zeilen

Übung 3.8

Ziel: Einüben von einfachen Bedingungen im WHERE; Verwendung von NULL; Anwenden von Datumsfunktionen

Aufgabenstellung:

- Geben Sie die Bestellnummer, den Wunschtermin, das Lieferdatum und die Liefer**verzögerung** aus.
- Ergebnisse von Bestellungen, die noch nicht geliefert wurden, sollen nicht ausgegeben werden.
- Geben Sie die beiden Datumsangaben (Lieferdatum und Wunschtermin) in einem leserlichen Format ohne Uhrzeit aus.

TIPPS:

- Beachten Sie, was das Start- und was das Enddatum ist! In diesem Fall kommt es auf die Formulierung der Aufgabenstellung an (siehe auch Lösungsteil).
- NULL kann nicht mit '=' oder '!=' abgefragt werden! Verwenden Sie IS NULL oder IS NOT NULL.

Results		Messages		
	OrderID	Lieferdatum	Wunschtermin	Lieferverzögerung
1	10248	16.07.1996	01.08.1996	-16
2	10249	10.07.1996	16.08.1996	-37
3	10250	12.07.1996	05.08.1996	-24
4	10251	15.07.1996	05.08.1996	-21
5	10252	11.07.1996	06.08.1996	-26

Ergebnis: 809 Zeilen

Modul 04 – Wildcards

Unter Wildcards verstehen wir in SQL Zeichen, die in der Suche als Platzhalter dienen, daher finden wir die Wildcards im WHERE.

Programmierer haben mit den Regular Expressions zwar deutlich mehr Möglichkeiten, werden hier aber eine gewisse Verwandtschaft erkennen.

% (Prozentzeichen) steht für beliebig viele unbekannte Zeichen (0 – n)

_ (Unterstrich) steht für **genau ein** unbekanntes Zeichen

[] (Eckige Klammern) stehen für **genau ein** unbekanntes Zeichen aus dem in den Klammern definierten Wertebereich

^ steht innerhalb der eckigen Klammern für NOT

| steht innerhalb der eckigen Klammern für OR

Übung 4.1

Ziel: Einüben der Verwendung von Wildcards

Aufgabenstellung:

- Geben Sie alle Produkte aus, deren Name mit 'L' beginnt.
- Wählen Sie passende Spalten aus.

	ProductID	ProductName
1	76	Lakkalikööri
2	67	Laughing Lumberjack Lager
3	74	Longlife Tofu
4	65	Louisiana Fiery Hot Pepper Sauce
5	66	Louisiana Hot Spiced Okra

Ergebnis: 5 Zeilen

Übung 4.2

Ziel: Einüben der Verwendung von Wildcards

Aufgabenstellung:

- Geben Sie alle Produkte aus, die ein 'ost' im Namen haben.
- Wählen Sie passende Spalten aus.

Results Messages		
	ProductID	ProductName
1	40	Boston Crab Meat
2	71	Flotemysost
3	33	Geitost
4	69	Gudbrandsdalsost
5	30	Nord-Ost Matjeshering

Ergebnis: 7 Zeilen

Übung 4.3

Ziel: Einüben der Verwendung von Wildcards

Aufgabenstellung:

- Geben Sie alle Produkte aus, deren Name mit D-L beginnt und mit a, b, c, d oder m, n, o endet.
- Wählen Sie passende Spalten aus.

Results Messages		
	ProductID	ProductName
1	31	Gorgonzola Telino
2	6	Grandma's Boysenberry Spread
3	24	Guaraná Fantástica
4	44	Gula Malacca
5	26	Gumbär Gummibärchen

Ergebnis: 8 Zeilen

Modul 05 – JOINS

Daten in der Datenbank sind in mehrere Tabellen aufgeteilt, um nach Möglichkeit Redundanz zu vermeiden. Viele Tabellen sind untereinander über sogenannte Schlüsselfelder miteinander verknüpft. Am Beispiel der Northwind-DB haben wir eine *Customers*-Tabelle und eine *Orders*-Tabelle. In der *Customers*-Tabelle gibt es eine *CustomerID*. Die *CustomerID* in der *Customers*-Tabelle ist der sogenannte *Primärschlüssel* oder *primary key*; von diesem sind alle anderen in der Tabelle abhängig, oder mit anderen Worten, wenn wir die Kundennummer kennen, können wir Kundeninformationen nachschlagen.

Auch in der *Orders*-Tabelle kommt die *CustomerID* vor, denn bei jeder Bestellung muss dabeistehen, welcher Kunde bestellt hat. Hier ist die *CustomerID* ein sogenannter *Fremdschlüssel* oder *foreign key*, also das Feld, über das eine Verknüpfung zur *Customers*-Tabelle besteht.

Brauchen wir nun die Kontaktinformationen oder den vollen Firmennamen des Kunden einer bestimmten Bestellung, verwenden wir einen sogenannten JOIN, um Zugriff auf Informationen von mehreren Tabellen zu haben.

Syntax:

```
SELECT  t1.Spaltenname1
        , t1.Spaltenname2
        , t2.Spaltenname1
FROM Tabellennamen1 AS t1 INNER JOIN Tabellennamen2 AS t2 ON t1.SpalteA = t2.SpalteA
```

Wir unterscheiden grob in INNER und OUTER JOIN; OUTER JOIN wird noch unterteilt in LEFT JOIN oder RIGHT JOIN. Wenn wir nur Daten aus Tabellen haben wollen, bei denen es Übereinstimmungen gibt, verwenden wir einen INNER JOIN. Wenn wir auch solche Daten haben wollen, wo es in der anderen Tabelle keine Übereinstimmungen dazu gibt, verwenden wir einen OUTER JOIN (also entweder LEFT JOIN oder RIGHT JOIN).

Anders ausgedrückt: Wenn wir nur die Kunden brauchen, die schon Bestellungen getätigt haben, verwenden wir einen INNER JOIN (denn hier gibt es eine Übereinstimmung in der *Orders*-Tabelle, weil bei jeder Bestellung dort auch die *CustomerID* eingetragen wird).

Wollen wir hingegen auch die Kunden ausgeben, die noch nichts bestellt haben, brauchen wir einen OUTER JOIN, denn für diese Kunden gibt es keine Übereinstimmung in der *Orders*-Tabelle, da ihre *CustomerID* dort erst eingetragen wird, wenn sie schon eine Bestellung getätigt haben.

Übung 5.1

Ziel: Einüben der Verwendung von JOINS

Aufgabenstellung:

- Geben Sie alle Kunden (Firmenname) aus Brasilien und deren Frachtkosten aus.

- Wählen Sie weitere nützliche Spalten, z.B. Bestellnummer und Kundennummer aus.

	Bestellnummer	Kundennummer	Firmenname	Frachtkosten
1	10250	HANAR	Hanari Carnes	65,83
2	10253	HANAR	Hanari Carnes	58,17
3	10256	WELLI	Wellington Importadora	13,97
4	10261	QUEDE	Que Delícia	3,05
5	10287	RICAR	Ricardo Adocicados	12,76

Ergebnis: 83 Zeilen

Übung 5.2

Ziel: Einüben der Verwendung von JOINS; Verwendung von Wildcards

Aufgabenstellung:

- Wir wollen bestimmte Produkte nachbestellen:
- Geben Sie die Namen der Anbieter (Supplier), die Sauce verkaufen, aus. (CompanyName, ProductName, Ansprechperson, Telefonnummer).
- Achtung: Wie heißt das Produkt/die Produkte? Verschiedene Möglichkeiten? Verschaffen Sie sich erst einen kleinen Überblick über die Produkte in der Datenbank und decken Sie mehrere Möglichkeiten ab.

	ProductName	CompanyName	ContactName	Phone
1	Northwoods Cranberry Sauce	Grandma Kelly's Homestead	Regina Murphy	(313) 555-5735
2	Louisiana Fiery Hot Pepper Sauce	New Orleans Cajun Delights	Shelley Burke	(100) 555-4822
3	Original Frankfurter grüne Soße	Plutzer Lebensmittelgroßmärkte AG	Martin Bein	(069) 992755

Ergebnis: je nach Abfrage 2 oder 3 Zeilen

Übung 5.3

Ziel: Einüben der Verwendung von JOINS; Verwendung von WHERE und IN; Verwendung von Serverfunktionen

Aufgabenstellung:

- Angenommen, es gab Beschwerden bei den Bestellungen 10251, 10280, 10990 und 11000.
- Welcher Angestellte hat diese Bestellungen bearbeitet?
- Geben Sie Vor- und Nachname in einem Feld als FullName aus.

Results		Messages
	OrderID	FullName
1	10251	Janet Leverling
2	10280	Andrew Fuller
3	10990	Andrew Fuller
4	11000	Andrew Fuller

Ergebnis: 4 Zeilen (nur die ausgewählten Bestellungen)

Übung 5.4

Ziel: Einüben der Verwendung von JOINS über mehrere Tabellen; Verwendung von WHERE und Wildcards

Aufgabenstellung:

- Welche Kunden haben Chai Tee gekauft und wieviel (einzelne Bestellungen)?
- Geben Sie OrderID, CustomerID, CompanyName, ProductName und Quantity an.

Results		Messages			
	OrderID	CustomerID	CompanyName	ProductName	Quantity
1	10285	QUICK	QUICK-Stop	Chai	45
2	10294	RATTC	Rattlesnake Canyon Grocery	Chai	18
3	10317	LONEP	Lonesome Pine Restaurant	Chai	20
4	10348	WANDK	Die Wandernde Kuh	Chai	15
5	10354	PERIC	Pericles Comidas clásicas	Chai	12

Ergebnis: 38 Zeilen

Übung 5.5

Ziel: Einüben der Verwendung von JOINS über mehrere Tabellen; Verwendung von WHERE und Wildcards; Verwendung von ORDER BY

Aufgabenstellung:

- Suchen Sie alle Bestellungen, bei denen Bier verkauft wurde. Welcher Kunde? Wieviel? Welches Bier?
- Der Produktname kann „Bier“ oder „Lager“ enthalten oder mit „Ale“ enden.
- Ordnen Sie die Ausgabe nach Menge (absteigend: größte zuerst) und Kundenname (alphabetisch).

	OrderID	CustomerID	CompanyName	ProductName	Quantity
1	10894	SAVEA	Save-a-lot Markets	Rhönbräu Klosterbier	120
2	11008	ERNSH	Ernst Handel	Sasquatch Ale	90
3	10990	ERNSH	Ernst Handel	Sasquatch Ale	60
4	10485	LINOD	LINO-Delicateses	Outback Lager	60
5	10326	BOLID	Bólido Comidas preparadas	Rhönbräu Klosterbier	50

Ergebnis: 114 Zeilen

Übung 5.6

Ziel: Üben von SELFJOIN; Verknüpfung einer Tabelle mit sich selbst

Aufgabenstellung:

- Wer ist der Chef von wem?
- Ausgabe: Name Angestellter, ID Angestellter, Name Chef, ID vom Chef
- Vergeben Sie neue Spaltenüberschriften
- Geben Sie den Vor- und Nachnamen in einer Spalte an
- Auch Employees, die keinen Vorgesetzten haben, sollen mit ausgegeben werden.

TIPPS:

- Für diese Profiübung müssen wir so tun, als gäbe es zwei Employee-Tabellen: Eine Tabelle für Mitarbeiter, eine für Vorgesetzte.
- In diesem Fall müssen die Tabellen ein ALIAS bekommen („AS name“).
- Die Spalte ReportsTo enthält die EmployeeID des Vorgesetzten.

	Mitarbeiter	Mitarbeiternummer	Vorgesetzter	ChefID
1	Nancy Davolio	1	Andrew Fuller	2
2	Andrew Fuller	2		NULL
3	Janet Leverling	3	Andrew Fuller	2
4	Margaret Peacock	4	Andrew Fuller	2
5	Steven Buchanan	5	Andrew Fuller	2
6	Michael Suyama	6	Steven Buchanan	5
7	Robert King	7	Steven Buchanan	5
8	Laura Callahan	8	Andrew Fuller	2
9	Anne Dodsworth	9	Steven Buchanan	5

Ergebnis: 9 Zeilen (Anzahl aller Employees)

Übung 5.7

Ziel: Üben von SELFJOIN; Verknüpfung einer Tabelle mit sich selbst

Aufgabenstellung:

- Angenommen, wir wollen eine Liste von Mitfahrgelegenheiten erstellen (wer lebt aller in derselben Stadt und könnte daher bei wem mitfahren?)
- Geben Sie alle Kunden aus, die in der gleichen Stadt wohnen, wie ein anderer Kunde (nicht sie selbst).
- Ausgabe: Customer1, City1, Customer2, City2
- Vergeben Sie Spaltenüberschriften
- Ordnen Sie die Ausgabe alphabetisch nach Städten und Kunden

TIPP: Die Spalte City wäre nur einmal notwendig, die doppelte Ausgabe dient nur der Kontrolle (und könnte später entfernt werden).

Results		Messages		
	Kunde	Stadt	Mitfahrgelegenheit	Kontrollfeld
1	Cactus Comidas para llevar	Buenos Aires	Océano Atlántico Ltda.	Buenos Aires
2	Cactus Comidas para llevar	Buenos Aires	Rancho grande	Buenos Aires
3	Océano Atlántico Ltda.	Buenos Aires	Cactus Comidas para llevar	Buenos Aires
4	Océano Atlántico Ltda.	Buenos Aires	Rancho grande	Buenos Aires
5	Rancho grande	Buenos Aires	Cactus Comidas para llevar	Buenos Aires

Ergebnis: 88 Zeilen

Modul 06 – DISTINCT, TOP, ORDER BY

Mit der Anweisung DISTINCT verhindern wir, dass Mehrfacheinträge ausgegeben werden.

Mit der Anweisung TOP können wir bestimmen, wie viele Zeilen ausgegeben werden sollen, z.B. nur die ersten 10, 100 oder 1000.

Damit wir sicherstellen können, was genau in diesen ersten n Zeilen steht, müssen wir auch ein ORDER BY verwenden.

Ohne weiteren Zusatz ordnet ORDER BY in alphabetischer Reihenfolge bzw. vom kleinsten zum größten Wert. Wollen wir in umgekehrter Reihenfolge ordnen, müssen wir DESC (descending) verwenden.

Sollen auch Ergebnisse angezeigt werden, die den gleichen Wert enthalten, wie in der letzten vom TOP-Befehl erfassten Zeile, dann verwenden wir die Anweisung WITH TIES.

Übung 6.1

Ziel: Einüben von TOP-Befehlen

Aufgabenstellung:

- Geben Sie mittels TOP-Befehl das teuerste Produkt aus
- Geben Sie mittels TOP-Befehl das günstigste Produkt aus (zweites SELECT-Statement)
- Wählen Sie in diesem Zusammenhang sinnvolle Spalten aus

Results		Messages	
	ProductID	ProductName	UnitPrice
1	38	Côte de Blaye	263,50

	ProductID	ProductName	UnitPrice
1	33	Geitost	2,50

Ergebnis: 2 x 1 Zeile

Übung 6.2

Ziel: Einüben von TOP-Befehlen; TOP PERCENT WITH TIES; JOIN

Aufgabenstellung:

- Suchen Sie die Top 10% der Produkte mit den größten Verkaufsmengen (ProductName, Quantity).

- Einschließlich Produkte mit der gleichen Einkaufsmenge wie das letzte in der ursprünglichen Ausgabe.

	ProductName	Quantity
1	Chartreuse verte	130
2	Wimmers gute Semmelknödel	130
3	Manjimup Dried Apples	120
4	Perth Pasties	120
5	Rhönbräu Klosterbier	120

Ergebnis: 234 Zeilen

Übung 6.3

Ziel: Einüben von TOP-Befehlen; Datumsformatierung; Serverfunktion

Aufgabenstellung:

- Geben Sie mittels TOP-Befehl die drei Mitarbeiter, die als erste eingestellt wurden, aus (die schon am längsten beim Unternehmen sind).
- Geben Sie zur Kontrolle das Einstellungsdatum (in einem sinnvollen Format) mit aus
- Geben Sie Vor- und Nachname in einer Spalte aus
- Wählen Sie nach Belieben weitere Spalten aus

	EmpName	Title	HireDate
1	Janet Leverling	Sales Representative	01.04.1992
2	Nancy Davolio	Sales Representative	01.05.1992
3	Robert King	Sales Representative	02.01.1994

Ergebnis: 3 Zeilen (es wurden nur die TOP 3 ausgewählt)

Modul 07 – Aggregatfunktionen; HAVING

Die häufigsten Aggregatfunktionen, die wir nutzen, sind MIN, MAX, COUNT (zählen), SUM (Summe) und AVG (Mittelwert).

Aggregatfunktionen werden für ganz konkrete Abfragen verwendet, nicht für einen möglichst großen Überblick über Spalten einer Tabelle.

Geben wir mehr als die Spalte in der Aggregatfunktion selbst im SELECT an, so müssen diese auch über GROUP BY gruppiert werden. Das GROUP BY bedeutet, die anderen Spalten sind ein „Pro“ etwas. Zum Beispiel der Durchschnitt der Frachtkosten pro Frachtunternehmen oder die Anzahl der Städte pro Land. Es kann auch nach mehreren Spalten gruppiert werden, zum Beispiel Anzahl Kunden pro Stadt und Land.

Vorsicht mit Gruppierungen: manche Spalten machen hier schlichtweg keinen Sinn. Die Summe der Frachtkosten pro Bestellung etwa wären wieder genau die ursprünglichen Frachtkosten, da sich aus dem einen Wert, der in jeder Bestellung vorkommt, keine Summe bilden lässt.

HAVING verwenden wir, wenn wir mit einem Wert vergleichen wollen, der erst durch eine Aggregatfunktion berechnet wird. Generell können wir uns aber merken: wenn es im WHERE möglich ist, dann im WHERE (schneller!); HAVING verwenden wir nur, wenn es sich nicht mit WHERE erledigen lässt.

Übung 7.1

Ziel: Üben von Aggregatfunktionen

Aufgabenstellung:

- Wie viele Produkte haben wir (=Northwind-Unternehmen) im Sortiment?
- Erstellen Sie eine Spaltenüberschrift.

Results		Messages	
		Anzahl Produkte	
1		77	

Ergebnis: Wir haben 77 Produkte im Sortiment.

Übung 7.2

Ziel: Üben von Aggregatfunktionen; DISTINCT

Aufgabenstellung:

- In wie vielen Ländern haben wir (=Northwind-Unternehmen) Kunden?
- Sorgen Sie dafür, dass die Ausgabe auch eine Spaltenüberschrift hat.

Results Messages	
	Länderanzahl
1	21

Ergebnis: Wir haben Kunden in 21 Ländern.

Übung 7.3

Ziel: Üben von Aggregatfunktionen; JOIN; GROUP BY; ORDER BY

Aufgabenstellung:

- Was ist der durchschnittliche Produktpreis pro Anbieter (Supplier)?
- Geben Sie Durchschnittspreis, SupplierID und CompanyName aus.
- Ordnen Sie die Ausgabe aufsteigend vom kleinsten zum größten Durchschnittspreis.

Results Messages			
	Durchschnittspreis/Anbieter	SupplierID	CompanyName
1	4,50	10	Refrescos Americanas LTDA
2	10,75	21	Lyngbysild
3	11,125	22	Zaanse Snoepfabriek
4	13,25	27	Escargots Nouveaux
5	14,025	19	New England Seafood Cannery

Ergebnis: 29 Zeilen

Übung 7.4

Ziel: Üben von Aggregatfunktionen; WHERE; GROUP BY; ORDER BY

Aufgabenstellung:

- Geben Sie die Summe der Frachtkosten pro Kunde im jeweiligen Land aus, die im Jahr 1996 entstanden sind.
- Erstellen Sie eine Spaltenüberschrift.
- Ordnen Sie die Frachtkosten absteigend nach der Summe der Frachtkosten.

Results Messages			
	Summe Frachtkosten/Kunde	CustomerID	ShipCountry
1	890,78	QUEEN	Brazil
2	816,33	QUICK	Germany
3	772,00	ERNSH	Austria
4	659,97	RATTC	USA
5	431,48	SPLIR	USA

Ergebnis: 67 Zeilen

Übung 7.5

Ziel: Üben von Aggregatfunktionen; HAVING; GROUP BY; ORDER BY

Aufgabenstellung:

- Wie viele Kunden gibt es pro Land?
- Es sollen nur die ausgegeben werden, wo mehr als 5 Kunden pro Land ansässig sind.
- In absteigender Reihenfolge geordnet (größte Anzahl zuerst).
- Ausgabe: Country, CustomerCount

Results Messages		
	CustomerCount	Country
1	13	USA
2	11	France
3	11	Germany
4	9	Brazil
5	7	UK

Ergebnis: 5 Zeilen

Übung 7.6

Ziel: Üben von Aggregatfunktionen; HAVING; GROUP BY; ORDER BY; JOINS; Serverfunktionen

Aufgabenstellung:

- Geben Sie die vollen Namen aller Angestellten aus, die mehr als 70 Bestellungen bearbeitet haben.
- Ordnen Sie die Ausgabe aufsteigend nach Anzahl der Bestellungen.

	FullName	NumberOfOrders
1	Robert King	72
2	Andrew Fuller	96
3	Laura Callahan	104
4	Nancy Davolio	123
5	Janet Leverling	127
6	Margaret Peacock	156

Ergebnis: 6 Zeilen

Übung 7.7

Ziel: Üben von Aggregatfunktionen; HAVING; GROUP BY; JOINS; IN

Aufgabenstellung:

- Haben die Angestellten Leverling und Peacock mehr als 100 Bestellungen bearbeitet?
- Geben Sie LastName und Anzahl der Bestellungen aus, aber nur, wenn sie über 100 Bestellungen bearbeitet haben.

	LastName	NumberOfOrders
1	Leverling	127
2	Peacock	156

Ergebnis: 2 Zeilen (beide haben über 100 Bestellungen geschafft)

Modul 08 – temporäre Tabellen, Views (Sichten), Prozeduren

Temporäre Tabellen ermöglichen uns, Informationen zwischenspeichern. Das können Inhalte ganzer Tabellen sein oder das von einem SELECT erzeugte Ergebnis, mit dem wir noch weiterarbeiten möchten.

Es gibt lokale und globale temporäre Tabellen; lokale temporäre Tabellen existieren nur in der Session, in der sie erstellt wurden, auf globale temporäre Tabellen können wir auch aus anderen Sessions zugreifen.

Beide existieren nur so lange, wie die Session, in der sie erstellt wurden, aufrecht ist.

Bei einer lokalen temporären Tabelle stellen wir ein #-Zeichen vor den Namen (#tablename), bei einer globalen temporären Tabelle zwei #-Zeichen (##tablename).

Views werden verwendet, um komplexere Abfragen, die häufiger gebraucht werden, nicht immer neu schreiben zu müssen, aber auch aus Sicherheitsgründen und um beispielsweise nur bestimmte Spalten der beteiligten Tabellen zur Verfügung zu stellen (Berechtigungen).

Eine View wird als virtuelle Tabelle verwendet; die Daten darin sind aktuell, denn bei Verwendung der View wird die dahinterliegende Abfrage ausgeführt.

Mit der View können wir wie mit einer Tabelle arbeiten; daraus ergeben sich sowohl Vor-, als auch Nachteile.

Eine Prozedur ist entfernt vergleichbar mit den Funktionen bei den Programmierern. Auch mit der Prozedur können wir Funktionen ausführen (Serverfunktionen oder UDF), auch die Prozedur rufen wir auf, so ähnlich wie beim function call beim Programmieren, und wie einer Funktion können wir auch der Prozedur Variablen übergeben.

Übung 8.1

Ziel: Erstellen einer View (Sicht); JOINS

Aufgabenstellung:

- Erstellen Sie eine View v_Rechnungsdaten.
- Wählen Sie sinnvolle Spalten aus mehreren Tabellen, die für die Ausgabe einer Rechnung notwendig wären.
- Führen Sie testweise ein SELECT * FROM v_Rechnungsdaten aus.

Ergebnis: 2155 Zeilen

Übung 8.2

Ziel: Erstellen einer Prozedur

Aufgabenstellung:

- Erstellen Sie eine Prozedur p_Customers_Cities.
- Beim Ausführen der Prozedur kann der Name einer Stadt übergeben werden, so dass nur die Kunden- und Bestelldaten der Kunden in dieser Stadt ausgegeben werden.
- Führen Sie die Prozedur für „Buenos Aires“ aus.
- Fortgeschrittenen-Zusatzaufgabe: Erstellen Sie eine temporäre Tabelle, fügen Sie das Ergebnis der Prozedur dort ein und ordnen Sie das Ergebnis nach OrderID

Results		Messages						
	CustomerID	CompanyName	PostalCode	Address	City	Country	OrderID	OrderDate
1	OCEAN	Océano Atlántico Ltda.	1010	Ing. Gustavo Moncada 8585 Piso 20-A	Buenos Aires	Argentina	10409	1997-01-09
2	RANCH	Rancho grande	1010	Av. del Libertador 900	Buenos Aires	Argentina	10448	1997-02-17
3	CACTU	Cactus Comidas para llevar	1010	Cerrito 333	Buenos Aires	Argentina	10521	1997-04-29
4	OCEAN	Océano Atlántico Ltda.	1010	Ing. Gustavo Moncada 8585 Piso 20-A	Buenos Aires	Argentina	10531	1997-05-08
5	RANCH	Rancho grande	1010	Av. del Libertador 900	Buenos Aires	Argentina	10716	1997-10-24

Ergebnis: 16 Zeilen

Übung 8.3

Bonusübung: Hierfür benötigen Sie auch das Modul 11 – CASE und das Modul 12 – CREATE,...

Wir verwenden die Tabelle, die wir im Modul 12 erstellen!

Ziel: Erstellen einer Prozedur; Üben von CASE

Aufgabenstellung:

- Erstellen Sie eine Stored Procedure p_Raise, die beim Ausführen das Gehalt aller Mitarbeiter, deren aktuelles Gehalt unter dem Durchschnitt liegt, um 20% erhöht, das der Mitarbeiter, deren aktuelles Gehalt über dem Durchschnitt liegt, um 10% erhöht, und das der Mitarbeiter, deren Gehalt 10.000 übersteigt, unverändert lässt.
- Führen Sie die Prozedur mehrmals aus und überprüfen Sie jedes Mal das Ergebnis, indem Sie das Gehalt der Mitarbeiter abfragen.

Results		Messages	
	EmployeeID	LastName	Salary
1	1	Davolio	3000,00
2	2	Fuller	8800,00
3	3	Leverling	2160,00
4	4	Peacock	5500,00
5	5	Buchanan	3840,00
6	6	Suyama	3720,00
7	7	King	2760,00
8	8	Callahan	3360,00
9	9	Dodsworth	3600,00

Ergebnis: 9 Zeilen (so viele, wie Mitarbeiter). Abhängig vom ursprünglichen Gehalt, das Sie vergeben haben, wird sich das Gehalt entsprechend ändern. Die Gehaltserhöhung ist deshalb so hoch, damit wir zum Testen rascher die 10.000er-Grenze erreichen können.

Modul 09 – SET-Operatoren

Die SET-Operatoren **UNION**, **UNION ALL**, **INTERSECT** und **EXCEPT** arbeiten mit Datensets, daher der Name. Wir können damit Listen, statt mehrerer Spalten, ausgeben, Vergleiche (z.B. vor und nach einem Update) anstellen und feststellen, welche Daten *nicht* übereinstimmen.

UNION macht zugleich auch ein DISTINCT: Mehrfacheinträge werden nicht ausgegeben. Möchte man alle Einträge ausgeben, auch doppelte, dann verwenden wir ein UNION ALL.

Da UNION ALL keine Vergleiche im Hintergrund anstellen muss (kein DISTINCT macht), ist es schneller als ein UNION. Wenn man also *mit Sicherheit* weiß, dass es keine Mehrfacheinträge geben kann (z.B., weil man mit IDs arbeitet), verwendet man ein UNION ALL, kein UNION, auch wenn man Mehrfachausgaben vermeiden will.

Bei der Verwendung von SET-Operatoren müssen wir darauf achten, dass wir in allen beteiligten SELECT-Statements die gleiche Spaltenanzahl verwenden und die Datentypen innerhalb einer Spalte kompatibel (bzw. implizit konvertierbar) sind.

Implizit konvertierbar heißt, dass der Server die Konvertierung für uns erledigen kann; z.B. von *varchar* nach *nvarchar*. Wenn wir aus irgendeinem Grund in einer Spalte z.B. einen String-Datentyp und den Datentyp *int* verwenden wollen, wäre das zwar möglich, wir müssten aber zuerst selbst (explizit) konvertieren (z.B. von *int* zu *varchar*).

Als Spaltenüberschrift wird der Spaltenname aus dem ersten SELECT verwendet, die nachfolgenden Spaltennamen überschreiben diesen nicht. Möchte man ein ALIAS vergeben, dann also im ersten SELECT.

Soll die Ausgabe mit ORDER BY geordnet werden, gilt ein ORDER BY für den gesamten Batch (alle beteiligten und mit UNION/ALL verknüpften SELECT-Statements).

Übung 9.1

Ziel: Üben von SET-Operatoren; UNION, UNION ALL; Datentypkonvertierung

Aufgabenstellung:

- Erstellen Sie eine Liste von allen Kontaktinformationen in der Northwind-DB: CompanyName, ContactName, ContactTitle und Phone von Kunden und Anbietern (Suppliers).
- Verwenden Sie dafür KEINE Joins, sondern UNION oder UNION ALL (überlegen Sie, was hier Sinn macht und warum).
- Erstellen Sie eine zusätzliche Spalte „Category“, in der jeweils eingetragen wird, ob es sich bei dem Kontakt um einen Kunden („C“ für Customer) oder Anbieter („S“ für Supplier) handelt.
- Testen Sie Ihr Ergebnis. Wenn alles soweit funktioniert, fügen sie auch noch eine Spalte „ContactID“ hinzu, in der die CustomerIDs und die SupplierIDs eingetragen werden.
- Testen Sie Ihr Ergebnis. Sobald dieser Abschnitt funktioniert, ordnen Sie die Ausgabe nach der ContactID, dann nach CompanyName und vergleichen Sie, wie sich das auf die Ausgabereihenfolge auswirkt.

	ContactID	CompanyName	ContactName	ContactTitle	Phone	Category
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	030-0074321	C
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	(5) 555-4729	C
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	(5) 555-3932	C
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	(171) 555-7788	C
5	18	Aux joyeux ecclésiastiques	Guyène Nodier	Sales Manager	(1) 03.83.00.68	S

Ergebnis: 120 Zeilen

Übung 9.2

Ziel: Üben von SET-Operatoren; TOP; Anwendungsbeispiel temporäre Tabellen

Aufgabenstellung:

- Geben Sie untereinander den höchsten und den niedrigsten Frachtkostenwert mit der dazugehörigen OrderID aus.
- Erstellen Sie eine zusätzliche Spalte, in der ein Vermerk steht („niedrigster“ bzw. „höchster Frachtkostenwert“).

TIPP: ORDER BY gilt für das gesamte UNION!

	OrderID	Freight	Anmerkung
1	10972	0,02	niedrigster Wert
2	10540	1007,64	höchster Wert

Ergebnis: 2 Zeilen

Modul 10 – Subqueries (Unterabfragen)

Eine Unterabfrage ist ein SELECT-Statement, das auch für sich allein stehen könnte, welches aber in ein anderes SELECT-Statement eingebaut wird, um die beiden Abfragen zu kombinieren. Wir möchten zum Beispiel nur die Bestellungen ausgeben, deren Frachtkosten über dem durchschnittlichen Frachtkostenwert liegt. Dazu müssen die durchschnittlichen Frachtkosten aber erst mittels Aggregatfunktion berechnet werden, damit wir sie im WHERE einbauen können. Wir können das über zwei Abfragen machen und den berechneten Wert einsetzen, oder wir können die Berechnung des Mittelwertes in eine Unterabfrage einbauen.

Subqueries oder Unterabfragen können im SELECT (wie eine Spalte), im FROM (wie eine Tabelle als Datenquelle) oder im WHERE verwendet werden.

Wird eine Subquery im FROM als Datenquelle verwendet, so muss ein Alias (in diesem Fall ein Tabellename) vergeben werden.

Übung 10.1

Ziel: Üben von Subqueries (Unterabfragen)

Aufgabenstellung:

- Geben Sie alle Bestellungen aus, deren Frachtkosten größer sind als der Durchschnitt der Frachtkosten.
- Ordnen Sie das Ergebnis absteigend (vom größten zum kleinsten Wert).

	OrderID	Freight
1	10540	1007,64
2	10372	890,78
3	11030	830,75
4	10691	810,05
5	10514	789,95

Ergebnis: 242 Zeilen

Übung 10.2

Ziel: Üben von Subqueries (Unterabfragen)

Aufgabenstellung:

Geben Sie die SupplierID, den CompanyName, die Kontaktinformation und das Land aller Supplier aus, die aus dem gleichen Land sind wie der Supplier Nr. 2.

Results		Messages		
	SupplierID	CompanyName	ContactName	Country
1	2	New Orleans Cajun Delights	Shelley Burke	USA
2	3	Grandma Kelly's Homestead	Regina Murphy	USA
3	16	Bigfoot Breweries	Cheryl Saylor	USA
4	19	New England Seafood Cannery	Robb Merchant	USA

Ergebnis: 4 Zeilen

Übung 10.3

Ziel: Üben von Subqueries (Unterabfragen); Serverfunktionen; Datumsfunktionen

Aufgabenstellung:

- Geben Sie die Namen und das Einstellungsdatum der Mitarbeiter aus, die im selben Jahr eingestellt wurden wie Mr. Robert King.
- Titel, Vorname und Nachname sollen überprüft werden.
- Uhrzeit soll nicht mit ausgegeben werden
- Soll Mr. Robert King selbst in der Ausgabe aufscheinen? Versuchen Sie beide Versionen, einmal mit, einmal ohne.

	FullName	HireDate
1	Mr. Robert King	02.01.1994
2	Ms. Laura Callahan	05.03.1994
3	Ms. Anne Dodsworth	15.11.1994

Ergebnis: 2 bzw. 3 Zeilen, abhängig davon, ob Mr. Robert King auch ausgegeben wird, oder nicht

Übung 10.4

Bonusübung: Hierfür benötigen Sie auch das Modul 12 – CREATE,...

Wir verwenden die Tabelle in der TestDB, die wir im Modul 12 erstellen!

Ziel: Üben von Subqueries (Unterabfragen); Serverfunktionen

Aufgabenstellung: Geben Sie die EmployeeID, den Vor- und Nachnamen (in einem Feld) und das Gehalt aller Mitarbeiter aus, die ein höheres Gehalt beziehen als der Mitarbeiter mit der EmployeeID 8.

Results		Messages	
	EmployeeID	FullName	Salary
1	2	Andrew Fuller	8000,00
2	4	Margaret Peacock	5000,00
3	5	Steven Buchanan	3200,00
4	6	Michael Suyama	3100,00
5	9	Anne Dodsworth	3000,00

Ergebnis: 5 Zeilen

Übung 10.5

Bonusübung: Hierfür benötigen Sie auch das Modul 12 – CREATE,...

Wir verwenden die Tabelle in der TestDB, die wir im Modul 12 erstellen!

Ziel: Üben von Subqueries (Unterabfragen); Serverfunktionen; IN

Aufgabenstellung: Geben Sie die EmployeeID, Name, Gehalt und Land der Mitarbeiter aus, die das niedrigste Gehalt in ihrem jeweiligen Land beziehen.

Results		Messages		
	EmployeeID	FullName	Salary	Country
1	3	Janet Leverling	1800,00	USA
2	7	Robert King	2300,00	UK

Ergebnis: 2 Zeilen (Mitarbeiter leben nur in UK und den USA)

Übung 10.6

Bonusübung: Hierfür benötigen Sie auch das Modul 12 – CREATE,...

Wir verwenden die Tabelle in der TestDB, die wir im Modul 12 erstellen!

Ziel: Üben von Subqueries (Unterabfragen); Serverfunktionen; BETWEEN

Aufgabenstellung:

Geben Sie die Namen der Employees aus, deren Gehalt zwischen dem niedrigsten Gehalt und 3000 liegt.
Ordnen Sie die Ausgabe aufsteigend nach Gehalt.

Results		Messages		
	EmployeeID	FullName	Salary	Country
1	3	Janet Leverling	1800,00	USA
2	7	Robert King	2300,00	UK
3	1	Nancy Davolio	2500,00	USA
4	8	Laura Callahan	2800,00	USA
5	9	Anne Dodsworth	3000,00	UK

Ergebnis: 5 Zeilen

Modul 11 – CASE

Mit CASE können wir ganz einfach unterschiedliche Möglichkeiten (Fälle) abdecken. Wenn – dann, nichts anderes macht CASE. In den meisten Fällen wird CASE in Verbindung mit Prozeduren und/oder Funktionen angewendet werden, aber wir können es z.B. auch ganz simpel im SELECT einbauen.

Übung 11.1

Ziel: Üben von CASE

Aufgabenstellung:

- Erstellen Sie im SELECT eine weitere Spalte, in die ein Kommentar eingetragen werden soll.
- Wenn es sich bei einem Land (in der Customers-Tabelle) um ein EU-Mitglied handelt, tragen Sie „EU“ ein.
- Wenn es sich um kein EU-Mitglied handelt, tragen Sie „nicht EU“ ein.
- Bei UK können Sie „Brexit“ eintragen.
- Für alle anderen tragen Sie „unbekannt“ ein.
- Wählen Sie nur einige Länder aus, Sie brauchen, um den Sinn dieser Übung zu erfüllen, nicht jedes einzelne Land anführen.
- Ordnen Sie die Ausgabe nach Zugehörigkeit
- Jedes Land soll nur einmal ausgegeben werden

Results Messages		
	Country	Zugehörigkeit
1	UK	Brexit
2	Austria	EU
3	France	EU
4	Germany	EU
5	Argentina	nicht EU

Ergebnis: 21 Zeilen

Übung 11.2

Ziel: Üben von CASE; Aggregatfunktionen; Datumsfunktionen; JOINS; Berechnungen

Aufgabenstellung erste Teilaufgabe:

- Erstellen Sie im SELECT eine weitere Spalte „Kundenstatus“.
- Hier soll dynamisch eingetragen werden, ob es sich um einen Premiumkunden oder einen Standardkunden handelt oder ob der Status unbekannt ist.
- Wenn die Einkaufssumme pro Kunde und Jahr mindestens 500 betragen hat, dann handelt es sich um einen Premiumkunden.

- Wenn die Einkaufssumme pro Kunde und Jahr weniger als 500 betragen hat, dann handelt es sich um einen Standardkunden.
- Andernfalls ist der Kundenstatus „unbekannt“.
- Berücksichtigen Sie für die Einkaufssumme den Discount. TIPP: Der Discount ist als Prozentwert (0-1) gespeichert; 0.25 bedeutet, es gab einen Discount von 25%.
- Überlegen Sie, ob Sie mit INNER oder OUTER JOIN arbeiten wollen und welchen Unterschied das für die Ausgabe macht!
- Geben Sie die KundenID, den Firmennamen, die Einkaufssumme (auf zwei Dezimalstellen gerundet), das Bestelljahr und den Kundenstatus aus geordnet nach Einkaufssumme aus.

Results Messages					
	Kundennummer	Firmenname	Bestelljahr	Einkaufssumme	Kundenstatus
1	FISSA	FISSA Fabrica Inter. Salchichas S.A.	NULL	NULL	unbekannt
2	PARIS	Paris spécialités	NULL	NULL	unbekannt
3	NORTS	North/South	1998	45.00	Standard
4	SPECD	Spécialités du monde	1997	52.35	Standard
5	FURIB	Furia Bacalhau e Frutos do Mar	1998	57.80	Standard

Ergebnis: mit INNER JOIN: 234 Zeilen; mit OUTER JOIN: 236 Zeilen

Aufgabenstellung zweite Teilaufgabe:

- Geben Sie die KundenID, Firmennamen, Einkaufssumme (auf zwei Dezimalstellen gerundet) und Bestelljahr (als Überprüfungsfeld) jeweils für die Jahre 1996, 1997 und 1998 geordnet nach der Einkaufssumme aus.

Ergebnis: 1996: 67 Zeilen; 1997: 86 Zeilen; 1998: 81 Zeilen

Aufgabenstellung dritte Teilaufgabe:

- Geben Sie die KundenID, Firmennamen, Einkaufssumme (auf zwei Dezimalstellen gerundet) und Bestelljahr (als Überprüfungsfeld) aus für die Fälle, wo kein Bestelljahr eingetragen ist.

Results Messages					
	Kundennummer	Firmenname	Bestelljahr	Einkaufssumme	Kundenstatus
1	FISSA	FISSA Fabrica Inter. Salchichas S.A.	NULL	NULL	unbekannt
2	PARIS	Paris spécialités	NULL	NULL	unbekannt

Ergebnis: 2 Zeilen

Modul 12 - CREATE, INSERT, UPDATE, DELETE

Wir können nicht nur Informationen aus der Datenbank abfragen, wir können auch Informationen in der Datenbank verändern oder überhaupt erst erstellen.

Mit CREATE erstellen wir eine Datenbank oder eine Tabelle (oder wie wir schon gesehen haben auch Views, Procedures und temporäre Tabellen).

Mit INSERT fügen wir Werte in die Spalten einer Tabelle ein.

Mit UPDATE verändern wir Werte in der Tabelle, z.B. fügen können wir damit einen neuen Kunden einfügen.

Mit DELETE können wir Informationen aus der Tabelle löschen (wenn es keine Abhängigkeiten gibt, die es verhindern).

Übung 12.1

Ziel: Erstellung von Mini-Übungsdatenbank mit einer Tabelle; Ergänzen einer Spalte; Einfügen von Werten

Aufgabenstellung:

- Erstellen Sie eine Testdatenbank (Name frei wählbar, z.B. TestDB)
- Erstellen Sie eine Kopie der Tabelle Employees aus der Northwind-DB in der TestDB.
- Fügen Sie eine Spalte „Salary“ hinzu.
- Fügen Sie (beliebige) Werte als Gehalt für die 9 Angestellten ein.
- Fragen Sie testweise EmployeeID, Name und Gehalt ab.

	EmployeeID	EmpName	Salary
1	1	Nancy Davolio	2500,00
2	2	Andrew Fuller	8000,00
3	3	Janet Leverling	1800,00
4	4	Margaret Peacock	5000,00
5	5	Steven Buchanan	3200,00
6	6	Michael Suyama	3100,00
7	7	Robert King	2300,00
8	8	Laura Callahan	2800,00
9	9	Anne Dodsworth	3000,00

Ergebnis: 9 Zeilen mit den eingefügten Werten als Gehalt